**Hands On : Create Microservices for account and loan**

**Account Microservice :**

**Steps :-**

1. **Project Initialization Using Spring Initializr -**

I created a new Spring Boot project using <https://start.spring.io>, configuring it with the following details:

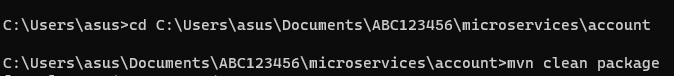
* **Group**: com.cognizant
* **Artifact**: account
* **Dependencies**: Spring Web, Spring Boot DevTools

1. **Project Setup and Build using Maven –**

After downloading the .zip file, I extracted it into my local workspace.

Then, I opened the terminal and navigated into the project directory.

To build the project, I ran the following Maven command:

****

**A black screen with white text

AI-generated content may be incorrect.**

1. **Importing Project into Eclipse -**

I imported the project into Eclipse via:

File → Import → Maven → Existing Maven Projects

1. **Implementing a controller method –**

I created a class ‘AccountController’ inside the package ‘com.cognizant.account.controller’ which contains controller method.

A screen shot of a computer program

AI-generated content may be incorrect.

1. **Running the Spring Boot Application –**

I executed the SpringLearnApplication.java class using:

Right-click → Run As → Java Application

The console confirmed that the Spring Boot app started successfully with embedded Tomcat running on port 8080.

**A screen shot of a computer

AI-generated content may be incorrect.**

**Output :**

This is the browser output after launching account project.

A screenshot of a computer

AI-generated content may be incorrect.

**Loan Microservice :**

**Steps :-**

1. **Project Initialization Using Spring Initializr -**

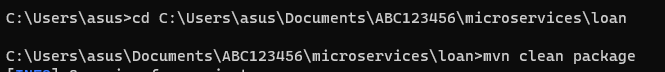
I created a new Spring Boot project using <https://start.spring.io>, configuring it with the following details:

* **Group**: com.cognizant
* **Artifact**: loan
* **Dependencies**: Spring Web, Spring Boot DevTools

1. **Project Setup and Build using Maven –**

After downloading the .zip file, I extracted it into my local workspace.

Then, I opened the terminal and navigated into the project directory. To build the project, I ran the following Maven command:



A black screen with white lines

AI-generated content may be incorrect.

1. **Importing Project into Eclipse –**

I imported the project into Eclipse via:

File → Import → Maven → Existing Maven Projects

1. **Implementing a controller method –**

I created a class ‘LoanController’ inside the package ‘com.cognizant.loan.controller’ which contains controller method.

**A screen shot of a computer program

AI-generated content may be incorrect.**

1. **Updating *application.properties* –**

We update application.properties port to 8081 as the default port 8080 is being used by ‘account’.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Running the Spring Boot Application –**

I executed the SpringLearnApplication.java class using:

Right-click → Run As → Java Application

The console confirmed that the Spring Boot app started successfully with embedded Tomcat running on port 8081.

A screenshot of a computer

AI-generated content may be incorrect.

**Output :**

This is the browser output after launching account project.

A screenshot of a computer

AI-generated content may be incorrect.